EN^DIB: User Controlled Editing

Invokes the Enter or Edit File Entries option of VA FileMan to edit records in a given file, allowing the user to select which fields to edit.

Input Variables

DIE (Required) The global root of the file in the form ^GLOBAL(or ^GLOBAL(# or the number of the file.

DIE("NO^") (Optional) Allows the programmer control of the use of the up-arrow in an edit session. If this variable does not exist, unrestricted use of the up-arrow for jumping and exiting is allowed.

The variable may be set to one of the following:

"**OUTOK**"	Allows exiting and prevents all jumping.
"**BACK**"	Allows jumping back to a previously edited field and does not allow exiting.
"**BACKOUTOK**"	Allows jumping back to a previously edited field and allows exiting.
"**Other value**"	Prevents all jumping and does not allow exiting.

DIDEL (Optional) Allows you to override the Delete Access on a file or subfile. Setting DIDEL equal to the number of the file before calling DIE allows the user to delete an entire entry from that file even if the user does not normally have the ability to delete. This variable does not override the DEL-nodes described in the Global File Structure chapter.

^DIC: Lookup

Given a lookup value, this entry point searches the requested Index(es) on a file and either finds a matching entry, adds an entry to the file, or returns a condition indicating that the lookup was unsuccessful.

See also IX^DIC and MIX^DIC1 for a comparison of how they each perform lookups.

Except for the DIC("W") variable, which is killed, the DIC input array is left unchanged by ^DIC.

Input Variables

DIC (Required) The file number or an explicit global root in the form ^GLOBAL(or ^GLOBAL(X,Y,.

DIC(0) (Optional) A string of alphabetic characters which alter how DIC responds. At a minimum this string must be set to null. A detailed description of these characters can be found later in this section, under DIC(0) Input Variables in Detail.

NOTE: If DIC(0) is null or undefined, no terminal output will be generated by the DIC routine.

The acceptable characters are:

Flag	Short Description
A	**A**sk the entry; if erroneous, ask again.
B	Only the **B** index is used when doing lookup to files pointed-to by starting file.
C	**C**ross-reference suppression is turned off.
E	**E**cho information.
F	**F**orget the lookup value.

I	**I**gnore the special lookup program.
K	Primary **K**ey is used as starting index for the lookup.
L	**L**earning a new entry is allowed.
M	**M**ultiple-index lookup allowed.
N	Internal **N**umber lookup allowed (but not forced).
O	**O**nly find one entry if it matches exactly.
Q	**Q**uestion erroneous input (with two ??).
S	**S**uppresses display of .01 (except B cross-reference match) and of any Primary Key fields.
T	Con**T**inue searching all indexes until user selects an entry or enters ^^ to get out.
U	**U**ntransformed lookup.
V	**V**erify that looked-up entry is OK.
X	E**X**act match required.
Z	**Z**ero node returned in Y(0) and external form in Y(0,0).

X If DIC(0) does not contain an A, then the variable X must be defined equal to the value you want to find in the requested Index(es).

If the lookup index is compound (i.e., has more than one data subscript), then X can be an array X(n) where "n" represents the position in the subscript. For example, if X(2) is defined, it will be used as the lookup value to match to the entries in the second subscript of the index. If only the lookup value X is passed, it will be assumed to be the lookup value for the first subscript in the index, X(1).

DIC("A") (Optional) A prompt that is displayed prior to the reading of the X input. If DIC("A") is not defined, the word Select, the name of the file, [i.e., $P(^GLOBAL(0),"^",1)], a space, the LABEL of the .01 field, and a colon will be displayed. If the file name is the same as the LABEL of the .01 field, then only the file name will be displayed. DIC(0) must contain an A for this prompt to be issued. For example, if the EMPLOYEE file had a .01 field with the LABEL of NAME, then FileMan would issue the following prompt:

```
Select EMPLOYEE NAME:
```

By setting DIC("A")="Enter Employee to edit: ", the prompt would be:

```
Enter Employee to edit:
```

Notice that it is necessary for the prompt in DIC("A") to include the colon and space at the end of the prompt if you want those to be displayed.

If the lookup index is compound (i.e., has more than one data subscript), then DIC("A") can be an array DIC("A",n) where "n" represents the position in the subscript. For example, DIC("A",2) will be used as the prompt for the second subscript in the index. If only the single prompt DIC("A") is passed, it will be assumed to be the prompt for the first subscript in the index DIC("A",1).

If DIC("A",n) is undefined for the "**n**th" subscript, then the "Lookup Prompt" field for that subscript from the INDEX file will be used as the prompt, or if it is null, the LABEL of the field from the data dictionary.

DIC("B") (Optional) The default answer which is presented to the user when the lookup prompt is issued. If a terminal user simply presses the Enter/Return key, the DIC("B") default value will be used, and returned in X. DIC("B") will only be used if it is non-null.

If the lookup index is compound (i.e., has more than one data subscript), then DIC("B") can be an array DIC("B",n) where "**n**" represents the position in the subscript. For example, DIC("B",2) will be used as the default answer for the prompt for the second subscript in

the index. If only the single default answer DIC("B") is passed, it will be assumed to be the default answer for the prompt for the first subscript in the index DIC("B",1).

DIC("DR") When calling DIC with LAYGO allowed, you can specify that a certain set of fields will be asked for in the case where the user enters a new entry. This list is specified by setting the variable DIC("DR") equal to a string that looks exactly like the DR string of fields that is specified when calling ^DIE. Such a list of what VA FileMan calls forced identifiers overrides any identifiers that would normally be requested for new entries in this file.

DIC("P") **NOTE:** As of Version 22 of FileMan, the developer is no longer required to set DIC("P"). The only exception to this is for a few files that are not structured like a normal FileMan file, where the first subscript of the data is variable in order to allow several different 'globals' to use the same DD. An example of this is the FileMan Audit files where the first subscript is the file number of the file being audited.

This variable is needed to successfully add the FIRST subentry to a multiple when the descriptor (or header) node of the multiple does not exist. In that situation, DIC("P") should be set equal to the subfile number and subfile specifier codes for the multiple. (See the File Header section of the Global File Structure chapter.) If the descriptor node for the multiple already exists, DIC("P") has no effect.

In order to automatically include any changes in the field's definition in DIC("P"), it is best to set this variable to the second ^-piece of the 0-node of the multiple field's definition in the DD. (See the Field Definition section of the Global File Structure chapter.)

Thus, for example, if file 16150 had a multiple field #9, set DIC("P") like this:

```
S DIC("P")=$P(^DD(16150,9,0),"^",2)
```

For more information, see Adding New Subentries to a Multiple below.

DIC("PTRIX",f,p,t)=d

DIC("PTRIX",f,p,t)=d where

f is the from (pointing) file number,

p is the pointer field number,

t is the pointed-to file number, and

d is an "^" delimited list of index names.

When doing a lookup using an index for a pointer or variable pointer field, this new array allows the user to pass a list of indexes that will be used when searching the pointed-to file for matches to the lookup value. For example, if your file (662001) has a pointer field (5) to file 200 (NEW PERSON), and you wanted the lookup on file 200 to be either by name ("B" index), or by the first letter of the last name concatenated with the last 4 digits of the social security number ("BS5" index): DIC("PTRIX",662001,5,200)="B^BS5". Note that if the call allows records to be added to a pointed-to file, then the list in the "PTRIX" entry should contain the "B" index. However, the "B" index would not need to be included in the list if the first index in the "PTRIX" array entry is a compound index whose first subscript is the .01 field.

DIC("S")

(Optional) DIC("S") is a string of M code that DIC executes to screen an entry from selection. DIC("S") must contain an IF statement to set the value of $T. Those entries that the IF sets as $T=0 will not be displayed or selectable. When the DIC("S") code is executed, the local variable Y is the internal number of the entry being screened and the M naked indicator is at the global level @(DIC_"Y,0)"). Therefore, to use the previous example again, if you wanted to find a male employee whose name begins with SMITH, you would:

```
S DIC="^EMP(",DIC(0)="QEZ",X="SMITH"
S DIC("S")="I $P(^(0),U,2)=""M"""
D ^DIC
```